OUR REF:
AN-C34/1

Greenbank Electronics

11/7/81

Tape UART Connections for new Software.

(Handwritten Notes)

Price 80p.

Introduction

This note describes the connections and circuit to configure a UART for tape. The two most recent pieces of firmware we have are ZYMON 1 and Tiny BASIC V5.02. These have both been written in accordance with the proposed memory map for future systems (described in our note ref. AN-C23/2, available from us at modest cost).

This UART circuit has been designed also with the same memory map in mind and so the arrangement described here is thus the one which is suggested for use with ZYMON 1 and Tiny BASIC V5.02.

The circuit diagram is shown on the last three pages of this note. Only the UART portion is shown, the actual interface to the tape recorder can be to the user's own design, or the 'Kemitron' TPA. tape interface card could be used. (The function of the tape interface is to process the serial data into a form which can be stored reliably on magnetic tape — very often for example the following scheme is used):

logic '1' : recorded as eight cycles of a 2400 Hz sinewave,

logic '0' : recorded as four cycles of a 1200 Hz sinewave.

The above system is known as 'Kansas City'

or 'CUTS', which is short for Cassette Users Tape Standard.

Circuit Design Considerations

A tape interface greatly adds to the facilities to be gained by installing the ZYMON 1 and/or Tiny BASIC V5.02 programs, but an ideal system should not make the use of a tape interface compulsary. In other words the system should be useable whether or not tape is available, and it is beneficial if the software can be informed somehow as to whether or not such an interface is connected.

The problem that could arise is that the software program would be trapped in a loop, waiting for data from a UART which would never appear if no UART was connected. (You might say why bother waiting, but it must remembered that the mode of sending data to and from tape is 'Asynchronous' in other words it can arrive at any time - it may be sooner or it may be later and so a long wait may be quite acceptable, it doesn't mean there's no UART there).

Two I/O Port addresses are allocated to the UART and the method used here to let the software know whether or not a tape interface circuit is available is to test 'Bit 5' of the 'Status' port address.

(Bit 5 is the 6th data line because the numbering begins 0, 1, 2 etc.)

The convention adopted for Bit 5 is as follows
BIT 5 = 1 means tape interface not available
BIT 5 = 0 " " " is available
It can often be convenient if a signal is given a name when it is being described in manuals and notes, and the name given to the Bit 5 signal is 'NAV' (short for '1' = 'not available'). 'NAV' is a signal to which reference is made in the ZYMON 1 Manual.

The polarity of NAV has been chosen deliberately so that physically withdrawing the tape UART and tape interface from the system will result in NAV being found to be in the 'not available' condition. (The reason for this is that an open-circuited line on the bus will usually be interpreted as a logic '1'. If it isn't then a pull up resistor (say 10k joining data line DB5 to +5V on the back board), will help ensure that it is.)

In short, if ZYMON 1 or Tiny BASIC V5.02 is to be run without a tape interface, it should work without any modifications to the system, but if not a 10k resistor on the bus could be tried.

Notwithstanding the above, it is strongly recommended that a tape interface be used to get the maximum use from ZYMON 1 or Tiny BASIC V5.01. The remainder of this leaflet will assume that this is to be employed.

The general requirements of the UART driving the Tape Interface are as follows:-

Saving Data

The software reads the UART status byte. As the UART is addressed as an I/O port it is an 'input' type of instruction, and the Port address for the UART status is as follows-

UART STATUS : Port Ø4

Only one bit is examined at this stage, this is bit 5, called 'NAV', as previously discussed. If it is '1' the software proceeds to ignore the UART and abandons the 'SAVE' operation (ZYMON 1 gives a 'D' error output - which means 'device not available').

However if the bit 5 (NAV) signal is 'Ø' then the presence of a Tape Interface is assumed and the data is saved via the UART in the normal way. (In case you don't know the 'normal way' this is briefly described next)

Bit 7 of the UART Status byte is examined next - this is TBMT, 'transmit buffer empty'. This is '1' to mean empty. If it is not '1' the Status byte is read again and again until TBMT is empty. (The UART is clocking data out of the 'SO' (serial out) pin continuously thus the transmit buffer is bound to be empty eventually - mostly in fact.)

Once the TBMT signal is found to be '1' the data which is to be saved is written into the buffer. This is done by outputting (writing) to the UART data port, in this system it is:

UART DATA = Port 05

This will immediately make the TBMT signal go to a '0' so when the status byte is read again (prior to saving a further byte of data) a delay may be needed until the transmit buffer is empty again. The delay is the result of the program 'looping' on the UART Status byte until TBMT goes to a '1' again.

The whole process repeats again and again until all the data to be saved has been saved.

Loading Data

The software for a load from tape first reads the status byte as ~~before~~ looking at bit 5, the NAV signal. As before it must be '0' for the load action to proceed, in the 'normal way', described as follows.

Serial data from the Tape Interface enters the UART SI pin, is stripped of some control bits ('start' and 'stop', and 'parity' if used, although 'parity' is not sent or received in this system). When a full eight bit byte is ready the UART sets the DAV 'Data available' flag (UART pin 19) high i.e. '1'.

The software is at this stage constantly examining this signal which is connected to bit 6 of the data bus when the UART status port address (Port Ø4) is being read. If the DAV signal is 'Ø' the status is read again and again until it is '1'. When it is '1' the data is loaded into the system by the execution of a 'read' or 'input' from the UART data port address (Port Ø5).

Whereas previously the TBMT flag was reset automatically by the UART (since the UART 'knew' when it had emptied the buffer) the DAV flag cannot be reset automatically by the UART (since the UART has no way of knowing directly when its data has been taken). The normal method of getting the DAV flag reset, ready for loading a new byte, is followed here. This is simply to join pins 4 and 18 on the UART. Pin 4 is taken low to enable the received data onto the data bus (pin 4 = $\overline{RDE}$), and this is connected to pin 18 ($\overline{RDAV}$) which resets the DAV flag at the same time.

Writing to the Status Port

Of course the Data Port receives both 'read' and 'write' signals because both operations are needed in a tape interface:

Read Data : Used when loading from tape.
Write Data : " " saving to "

We have already seen how the UART Status Port has to be read to determine the 'status'

of the various buffers and flags e.g. TBMT and DAV. It however has not yet been made clear what function writing to the status PORT can have. On the face of it, it is a useless function as the UART always knows its own status – it never needs to be told!

On some systems 'writing to the status port' is used to select different baud rates or to program various UART data formats, but this is not used as a technique in this system. The baud rate is fixed, not programmable (at least not implicitly by ZYMON 1 or Tiny BASIC V5.02) and the data formats are preset by a DIL switch or wire links.

As an aside the 'data formats' to which we refer are the ones which are set by the UART pins 34–39, a brief description of the settings used here follows:

Pin No.	Signal	Level	Description of setting used.
Pin 34	CS	'1'	To enable the settings on pins 35–39 to be entered into the UART control register
35	NP	'1'	For 'no parity'
36	TSB	'1'	To transmit two stop bits.
37	NB1	'1'	Both '1' means that each character will contain 8 bits (what we want).
38	NB2	'1'	
39	EPS	'1'	To give 'even' parity but note as pin 35 is '1' the level on this pin is immaterial – no parity is transmitted or received.

So much for what some systems do with the function of 'writing to the status port' – what do we do with it?

Early users of Tiny BASIC (Version V1.1 or V1.2) will remember that writing to the status was used as a software reset, to reset the UART. Since those early days a reset line has been defined on the ISBUS back board ("A" side, pin 21: NRST), and this is used to reset the UART now. Thus ZYMON 1 and Tiny BASIC V5.02 no longer write automatically to the status byte when saving data.

It will be remembered that a signal 'NAV', connected to bit 5 of the data bus, has been defined. The circuit diagram showing the UART connections also shows a front panel switch which can force NAV to '1' (Interface 'OFF' or not available) or to a '0' (interface 'ON' or 'available'); an LED indicator is lit when the NAV signal is '0' (meaning interface 'ON').

Also is shown an 'AUTO' position for the switch. When the switch is in this position the state of NAV can be set by software, so that for example some things can be recorded on tape and some can be omitted (automatic censorship of rude words for example!)

The method used to set or reset the NAV line is to write specific data to the UART status port address. Specifically, the data should contain a '1' in bit 5 of the data byte to

set NAV to '1', and a 'Ø' to reset NAV to 'Ø'.

If nothing else the ability to turn a light on and off will confirm that the UART addressing is correct, and would make a useful 'novelty' light for say 'reaction timer' programs or 'red alert - space invaders present'.

Of course it does not matter what levels are used on the other bits (... i.e. other than bit 5), but it would be best if they were left undisturbed in case somebody else had some bright idea for using one or other of them as well).

Applications of the technique

It must be admitted that there is not much use in having an 'on - auto - off' switch for tape. The main point in suggesting it is to pull all UARTs in a system into line, so the same software can be used to drive each. The Tiny BASIC V5.Ø2 program has limited facilities for driving a printer (at ports Ø6 and Ø7) and an on-auto-off switch is invaluable there - together with the ability of the program to ignore the printer if there isn't one available. (It must be remembered not everyone has a printer!)

One problem the switch would solve is the danger of corrupting a Tiny BASIC program by playing a tape which has some other programs on it while the tape recorder is still connected. Now the interface can now simply be turned 'OFF'.

Simplifications to Circuit Shown

There are a number of simplifications possible if an interface is being wired specifically for the 'Tiny BASIC V5.02/ZYMON 1' application.

① The T1-T6 inputs of the 8131 can be wired directly to 0V or +5V as appropriate, there is no need for the DIL Switch and pull-up resistor arrangement if the ports are not being changed regularly to different addresses.

② On a lightly loaded bus the two 74LS125 buffers for NWDS and NRDS are not absolutely essential; they are included here only to reduce the loading to 1 LS load

③ Similarly the 74LS245 may not be needed to buffer the data bus in some small systems.

④ As noted on the UART diagram pull ups on pins 34-39 are not needed with some common UARTs. If these pins were wired directly to 0V or +5V then the pull-up resistors are not needed at all for any type of UART.

⑤ If the 'on-auto-off' switch is not needed it can be left out, similarly the LED indicator and transistor. NAV can be set to the desired 'Ø' level by omitting the 74LS74 and jamming pin 9 of the 74LS125 shown to 0V. If extreme economy is desired the 74LS125 can be left out and the necessary Ø forced onto NAV by connecting a diode from $\overline{SWE}$, UART

pin 16 to bit 5 of the UART data bus

e.g UART pin 16 ——|◀|—— UART pin 7 and/or 31

(—|▭|—)

The diode is not as good as the 'Ø' output of the 74LS125 tristate buffer it replaces - a low forward drop Schottky Diode or Germanium diode should be used but an ordinary 1N4000 series or 1N4148 has reportedly given adequate results.

Tailpiece

If this circuit is being adopted by an existing user of the Kemitron TPA board, he may care to seize the opportunity to make a small modification to this as well. The reason for the modification is to improve the long term stability of the master oscillator on the TPA2 board.

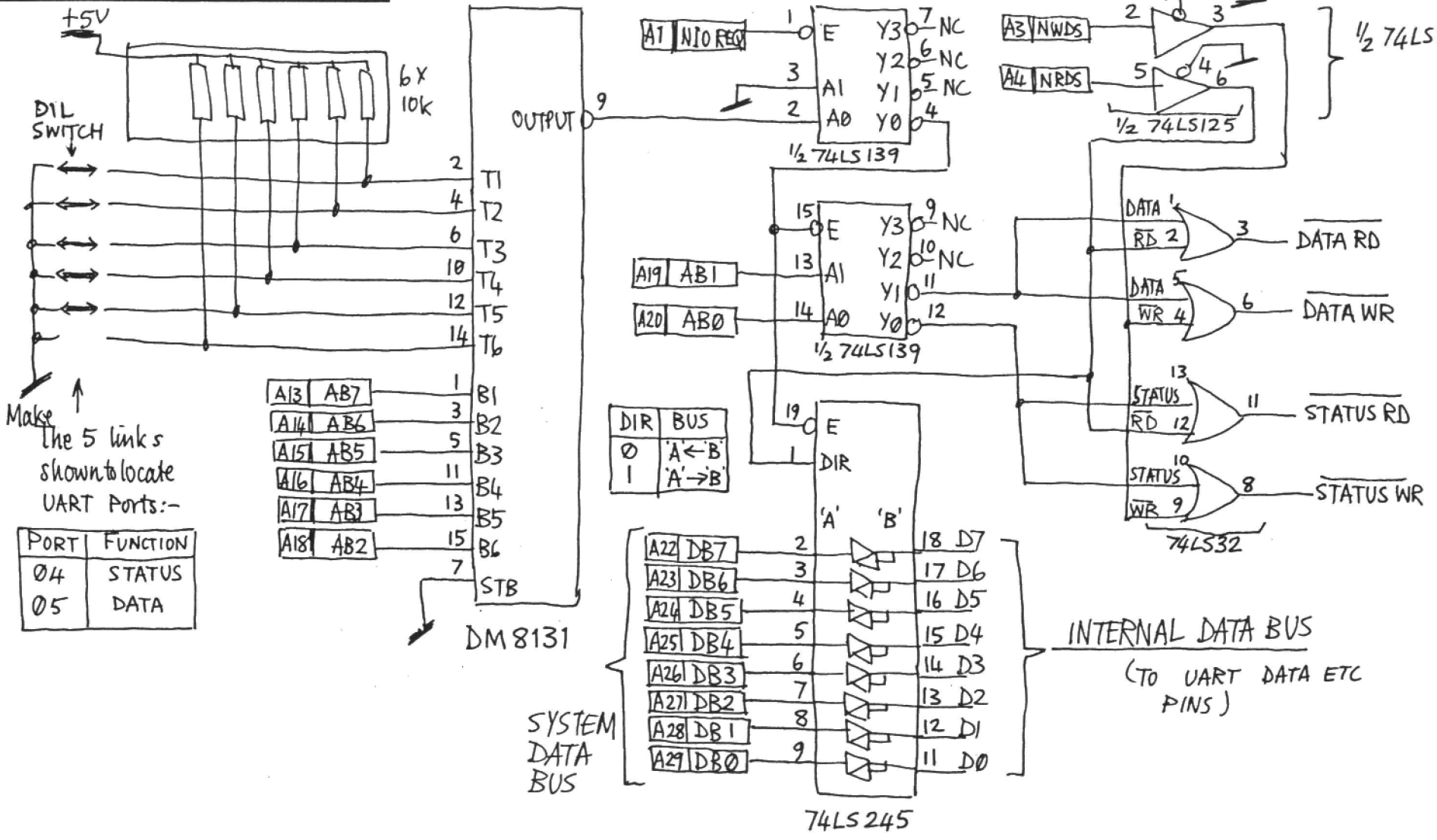
The device used was a '555' precision timer, which sounds ideal, but we have some reservations about how 'precise' a 'precision timer' actually is. Our own feelings are that the 555 is not equal to the task, and some more stable alternative is to be preferred. An arrangement of two 74LS series monostables, arranged as an oscillator has reportedly given improved results, and of course a crystal controlled oscillator and divider circuit would be ideal. The '555' replacement should have a frequency of 19.2 kHz.

Do not sacrifice the 555 altogether - you will need to read your old tapes at least one more time, and an adjustable baud rate is necessary to read tapes made on a machine

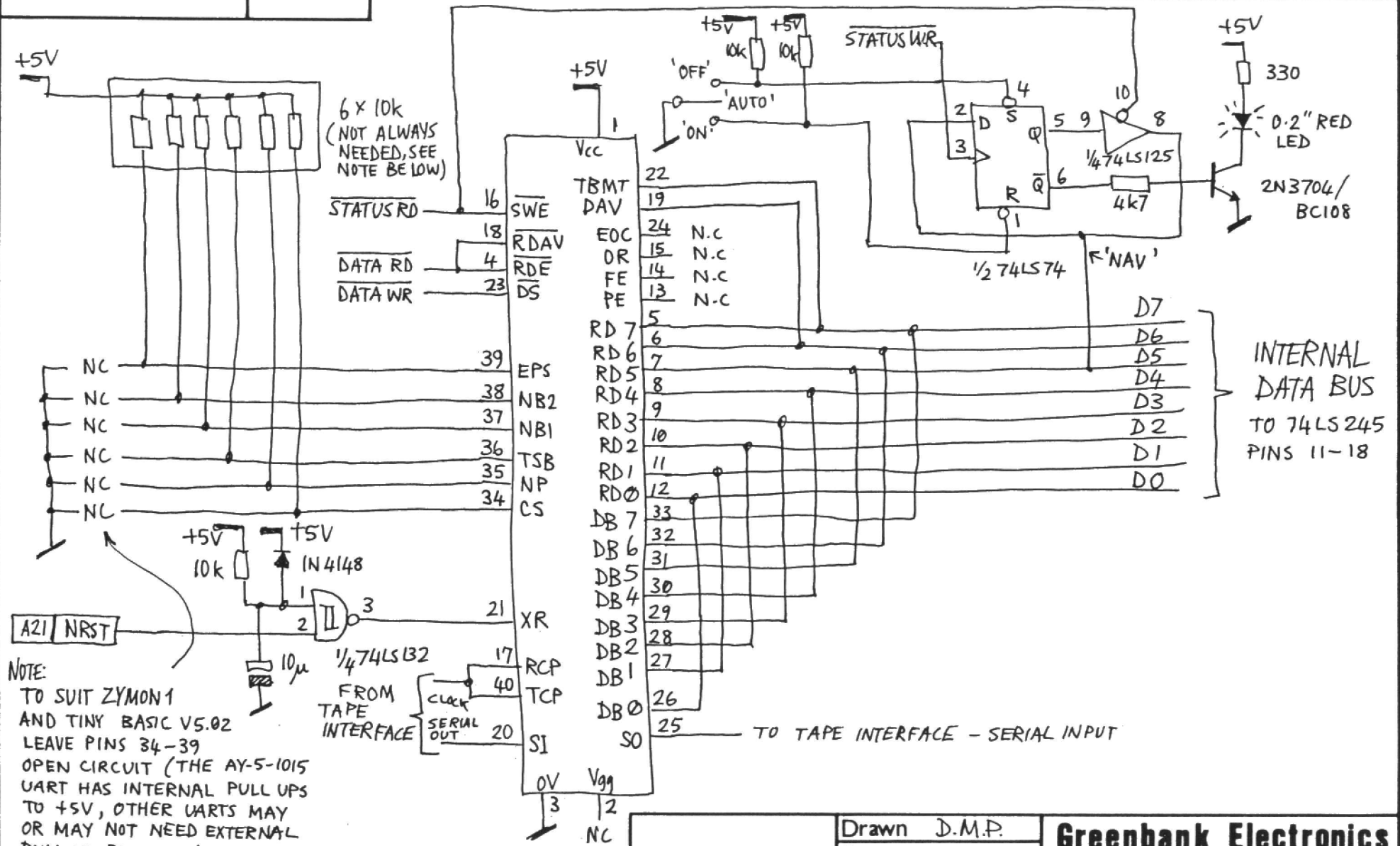
running at a slightly different speed to your own (e.g. pre-recorded software).

If the 555 is causing you problems it will exhibit the following symptoms: Tapes made will load fine immediately after they are made, but will give corruptions a day or so later if the computer is hotter or colder than it was when the tape was made.

DMP. 11-7-81.

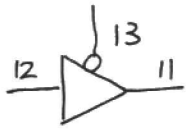


	Drawn D MP	**Greenbank Electronics**
	Date 10-7-81	TAPE INTERFACE UART CIRCUIT DIAGRAM
	Scale —	1

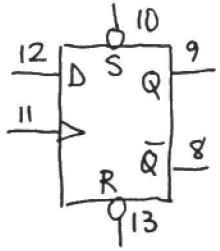


	Drawn D.M.P.	**Greenbank Electronics**
	Date 10-7-81	TAPE INTERFACE UART CIRCUIT DIAGRAM
	Scale –	2

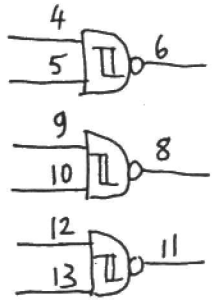
UNUSED FUNCTIONS



1/4 74LS125



1/2 74LS74



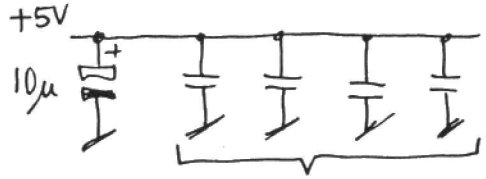
3/4 74LS132

UNUSED FUNCTIONS

POWER SUPPLIES

DEVICE	+5V	0V
74LS32	14	7
74LS74	14	7
74LS125	14	7
74LS132	14	7
74LS139	16	8
DM8131	16	8
AY-5-1015	1	3

DECOUPLING CAPACITORS



4 x 100nF DISTRIBUTED ROUND BOARD

	Drawn D.M.P.	**Greenbank Electronics**
	Date 10-7-81	TAPE INTERFACE UART CIRCUIT DIAGRAM
	Scale —	3

91